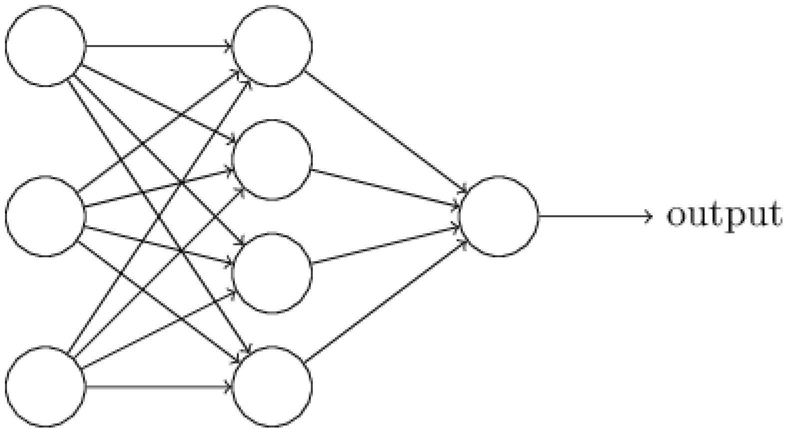
Neural Networks - II

Machine Learning – CSE 543 / ECE 563

Terminology

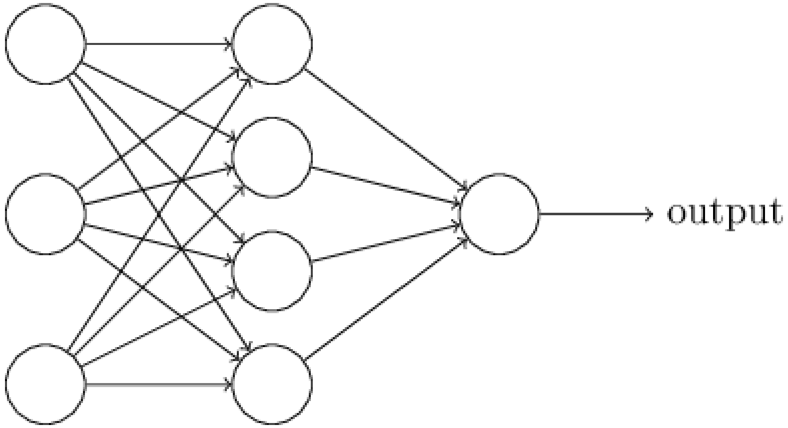
- Neurons or nodes
  - Each layer comprises of one or more neurons or nodes
  - Each neuron is responsible for a simple computation
    - Inner product + activation (linear or nonlinear)
- Input layer
  - Neurons take input values (often ‘raw data’, e.g., pixels of an image)
- Output layer
  - Neurons that output the predictions (class label, regressed value, rank, etc.)
- Hidden layer
  - All intermediate layers between input and output
- Computational Graphs
  - A way to represent neural networks
  - Computation at each node

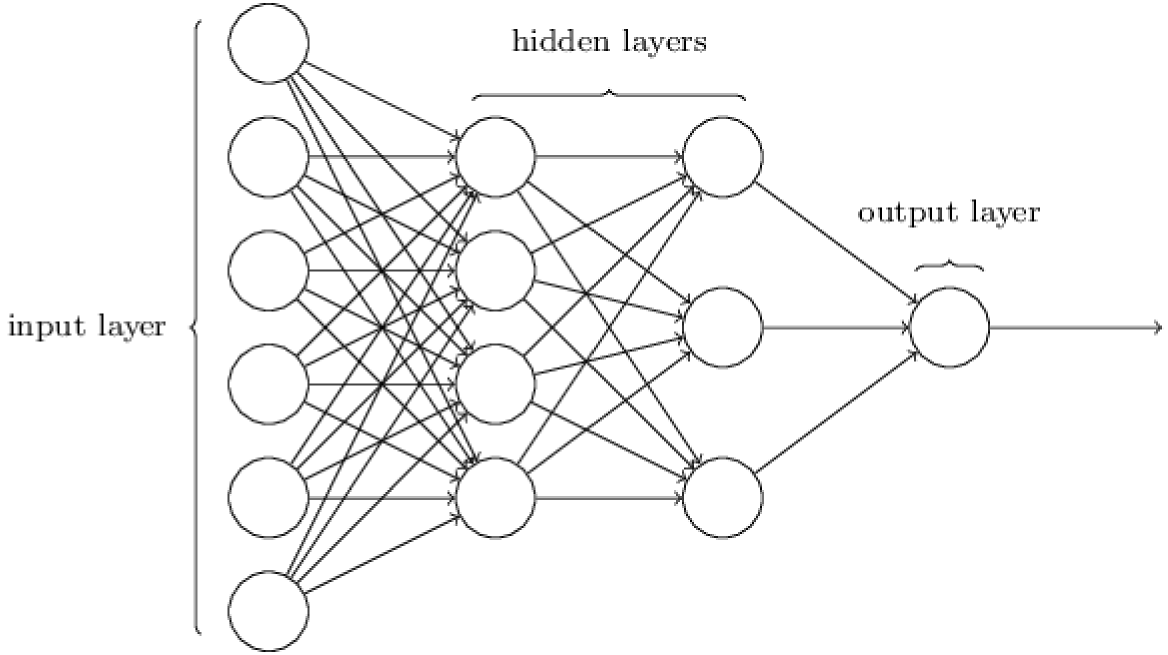
Feed Forward Neural Networks (aka Multi-Layer Perceptron)

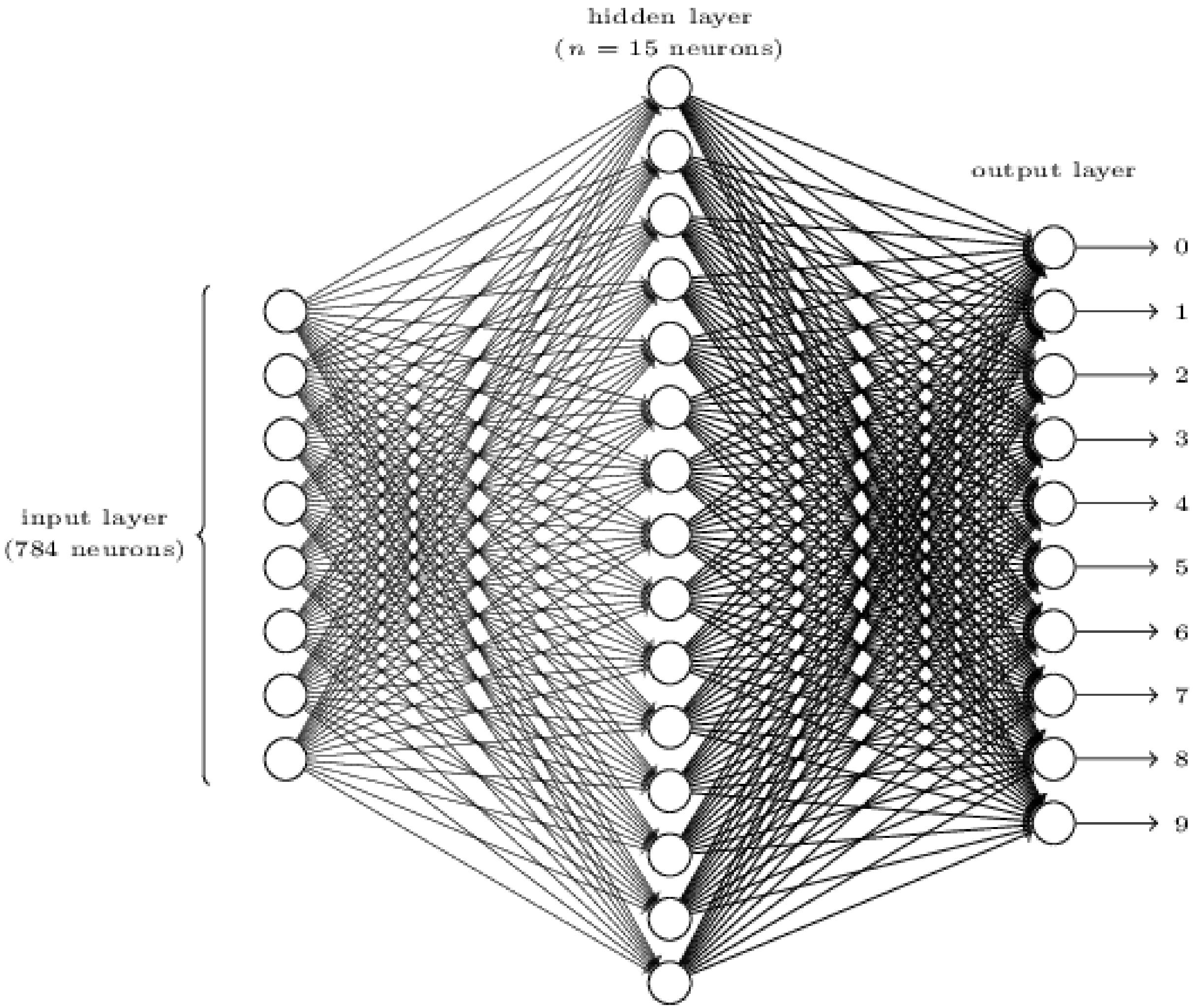


- Feedforward neural nets
  - Information flows in the forward direction (from input layers toward output layers)
  - Output of a node only connects to nodes in the 'next' layer
    - Never to a node in the same layer
    - Never to a node in the previous layer
- Recurrent neural networks (RNNs) are <u>not</u> feedforward networks

Architecture of Neural Nets

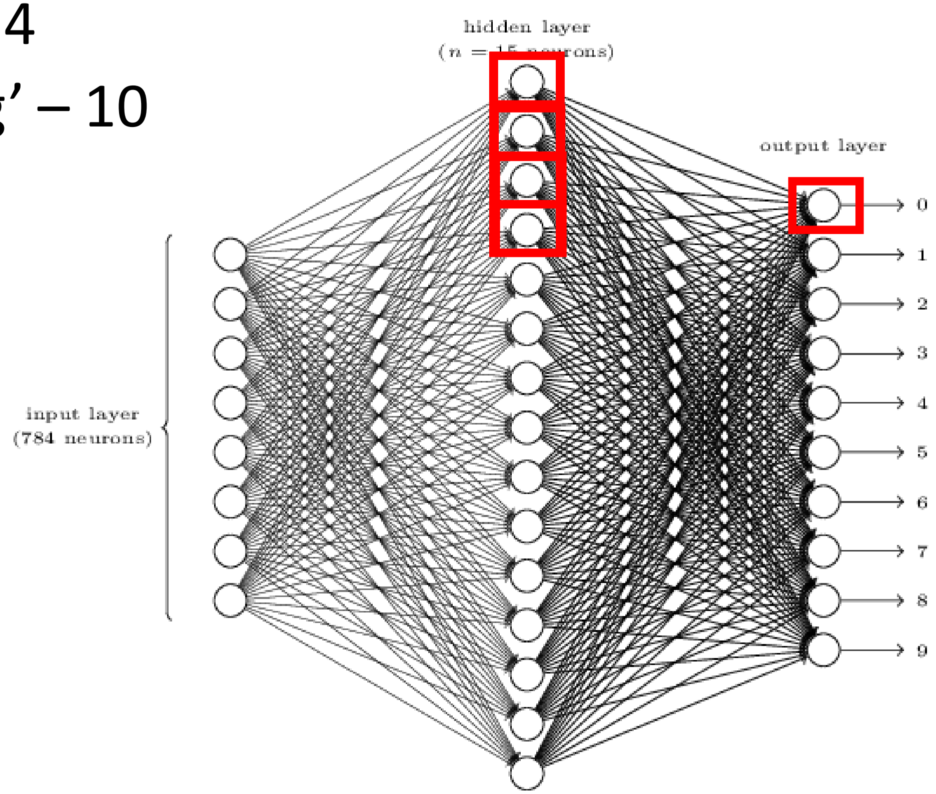


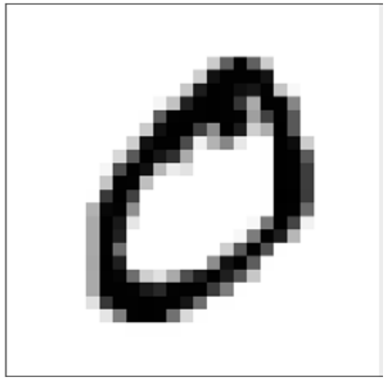




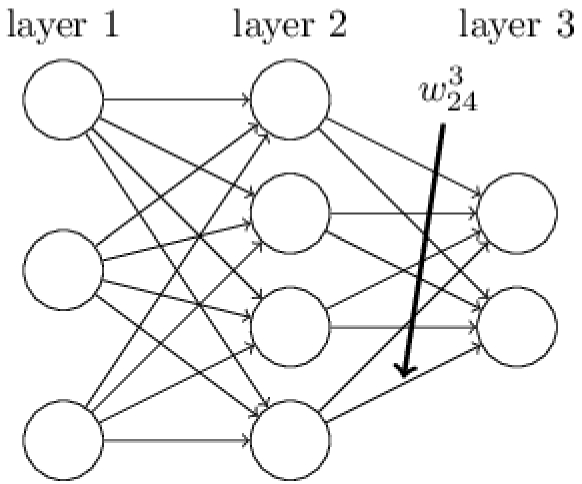
Output Layers

- Classifying digits (10 class problem)
  - How many output nodes?
  - Binary encoding – 4
  - 'one-hot-encoding' – 10





Matrix Notation for Neural Nets



w^l_{jk} is the weight from the k^{th} neuron in the $(l-1)^{\text{th}}$ layer to the j^{th} neuron in the l^{th} layer

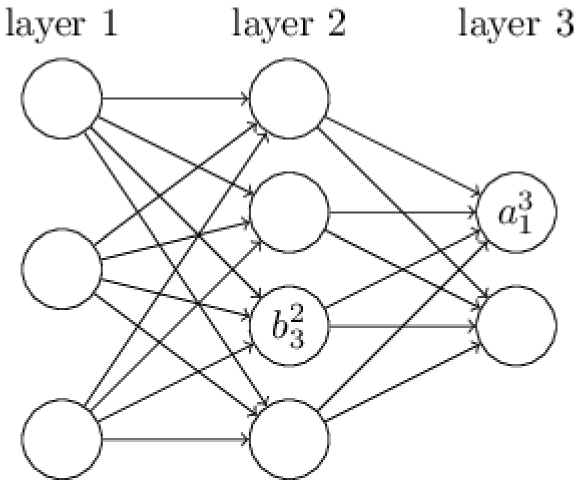
$$w^l \in \mathbb{R}^{n_l \times n_{l-1}} \quad \text{(weight matrix)}$$

$$b^l \in \mathbb{R}^{n_l} \quad \text{(vector of biases, often included in } w^l\text{)}$$

$$a^l \in \mathbb{R}^{n_l} \quad \text{(vector of activations)}$$

$$z^l \in \mathbb{R}^{n_l} \quad \text{(vector of weighted inputs)}$$

Goal: learn weights *w* and biases *b* for each layer such that predicted labels are consistent with the training data



$$a^l_j = \sigma\left(\sum_k w^l_{jk} a^{l-1}_k + b^l_j\right)$$

$$\sigma\left(\begin{bmatrix} 2 \\ 3 \end{bmatrix}\right) = \begin{bmatrix} \sigma(2) \\ \sigma(3) \end{bmatrix} = \begin{bmatrix} \frac{1}{1+e^{-2}} \\ \frac{1}{1+e^{-3}} \end{bmatrix}$$

Activations applied element-wise

Can be written in vector notation

$$a^l = \sigma(w^l a^{l-1} + b^l) = \sigma(z^l)$$

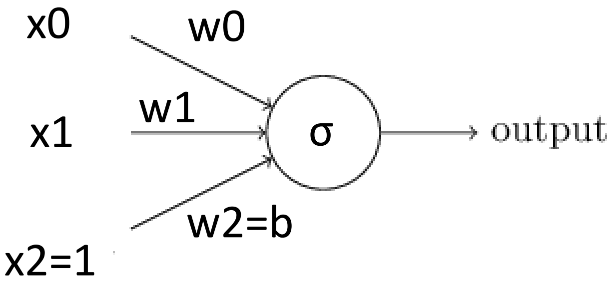
where, the vector $z^l = w^l a^{l-1} + b^l$

has elements $z^l_j = \sum_k w^l_{jk} a^{l-1}_k + b^l_j$

For sigmoid activation, z^l_j are also called as logits

Forward Pass

- A simple architecture
  - Representing this network as a computational graph
  - Let true y = 1, and output layer activation a^L

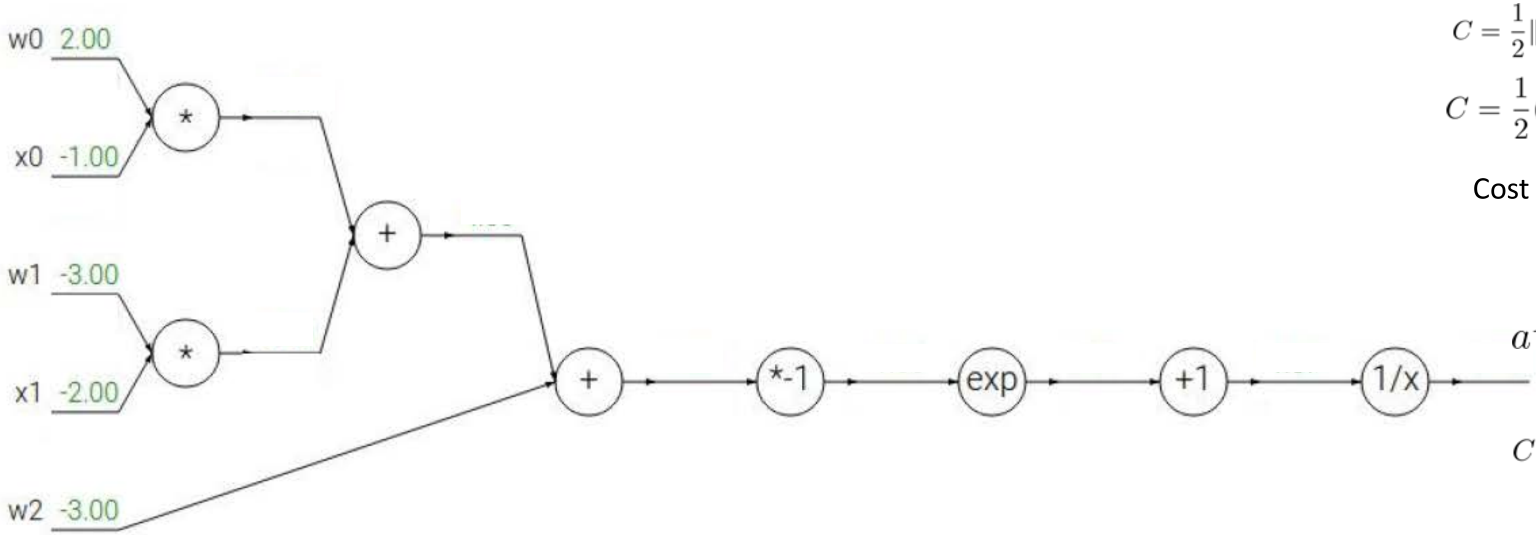


$$a^L = f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$C = \frac{1}{2}\|y - a^L\|^2 = \frac{1}{2}\sum_j (y_j - a_j^L)^2;$$

$$C = \frac{1}{2}(y - a^L)^2 = \frac{(1-0.73)^2}{2}$$

Cost for one training sample

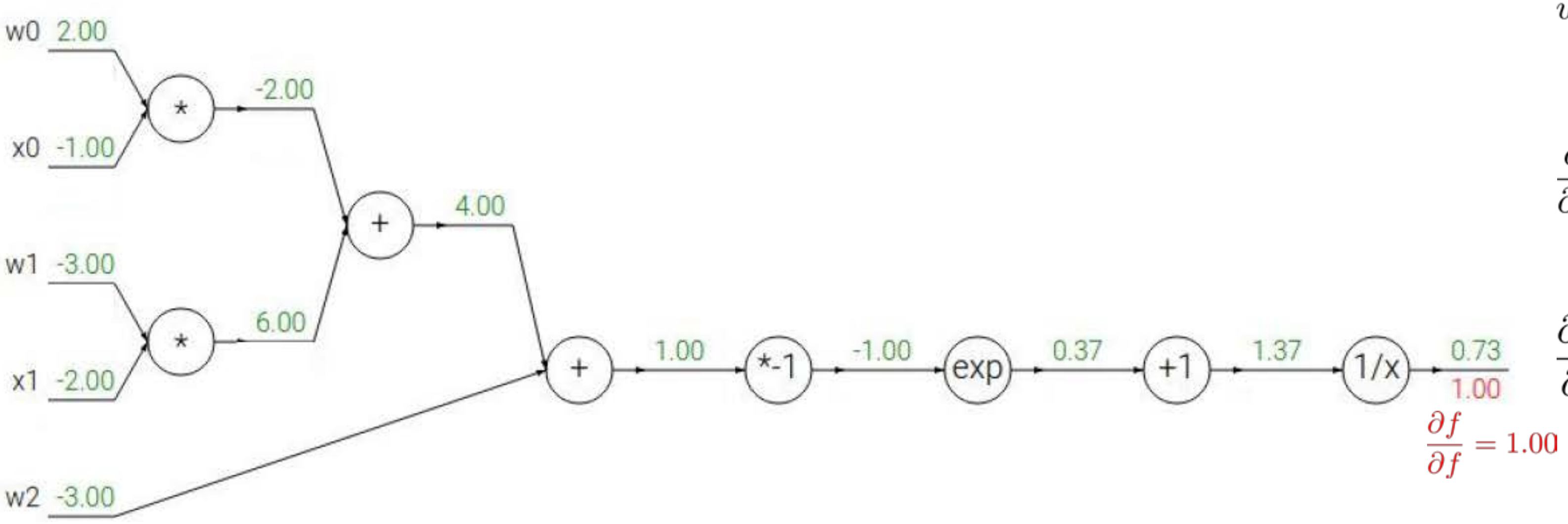


$$a^L = 0.73$$

$$C = \frac{1}{2n}\sum_x \|y(x) - a^L(x)\|^2;$$

Cost averaged over all training samples

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



Gradient update rule

$$w_j \leftarrow w_j - \eta \frac{\partial C}{\partial w_j}$$

Chain rule

$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial f(w,x)} \frac{\partial f(w,x)}{\partial w_j}$$

$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

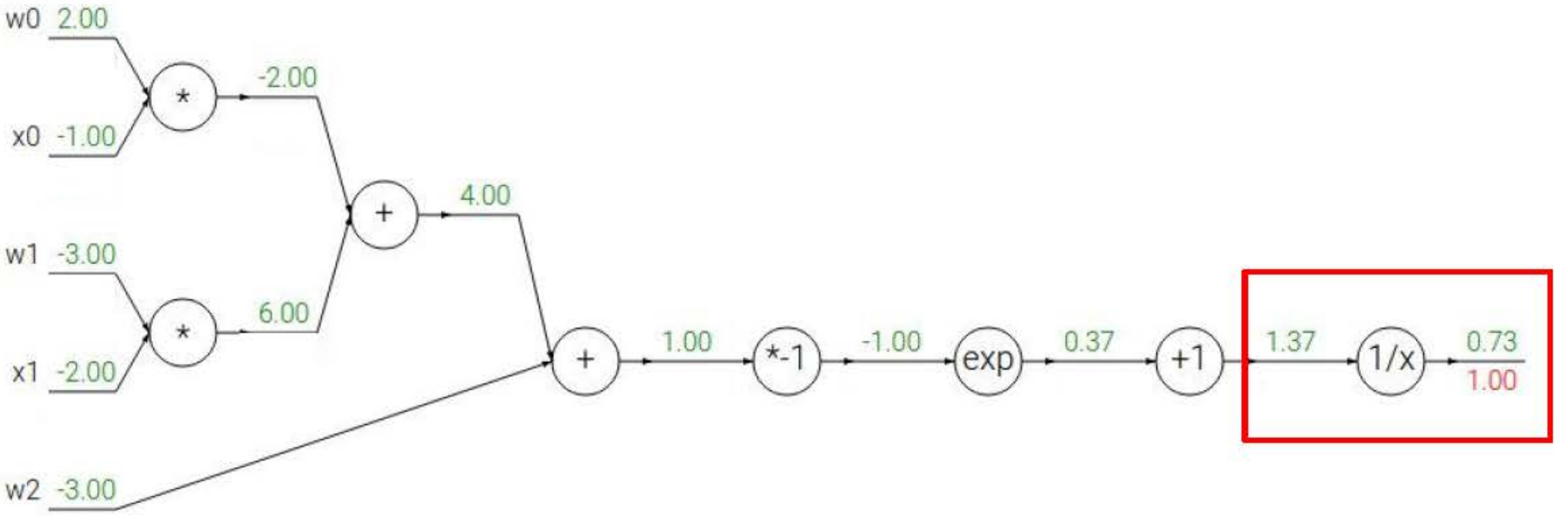
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

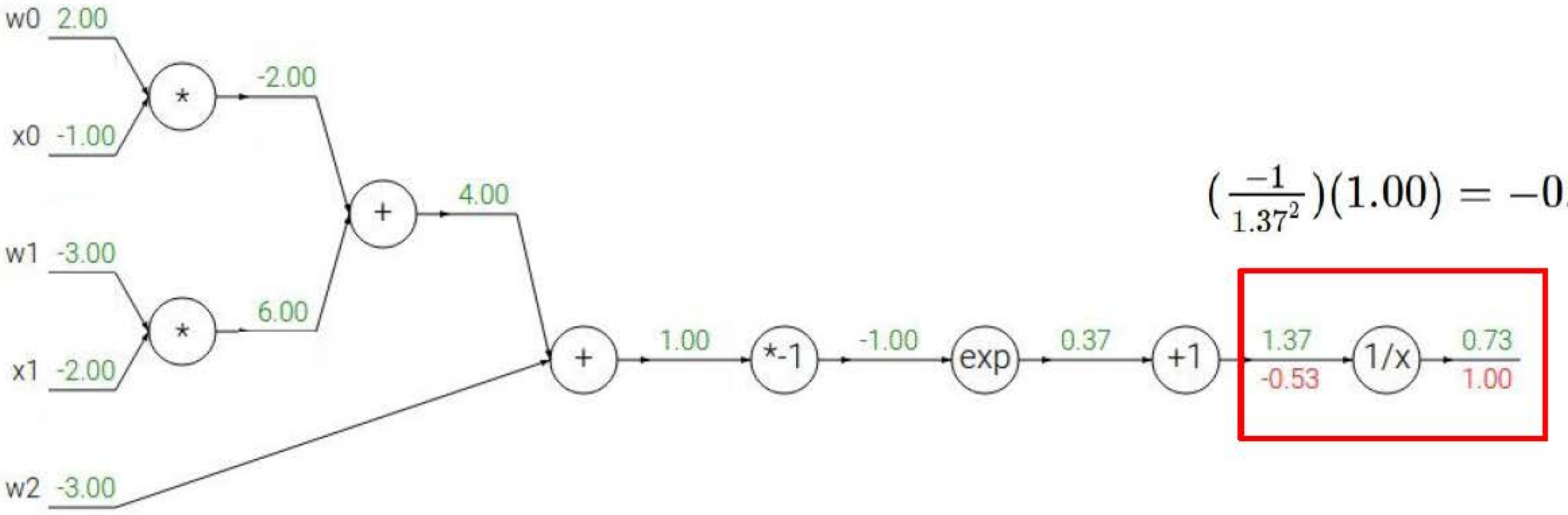
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\left(\frac{-1}{1.37^2}\right)(1.00) = -0.53$$

$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

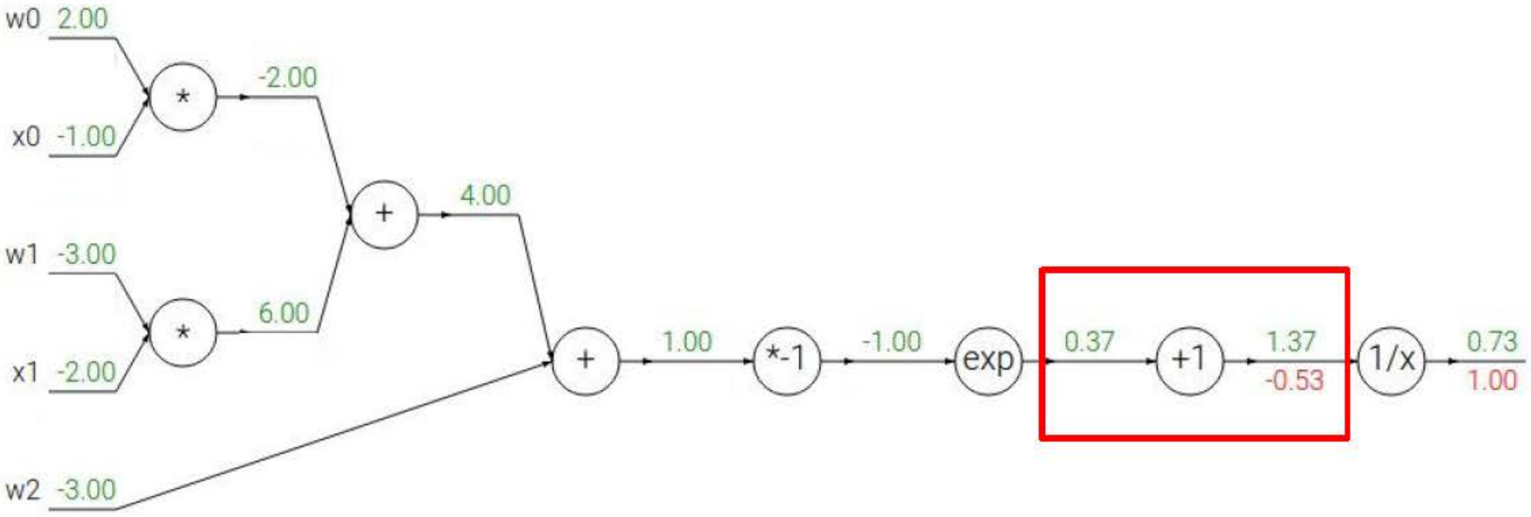
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$\boxed{f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2}$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

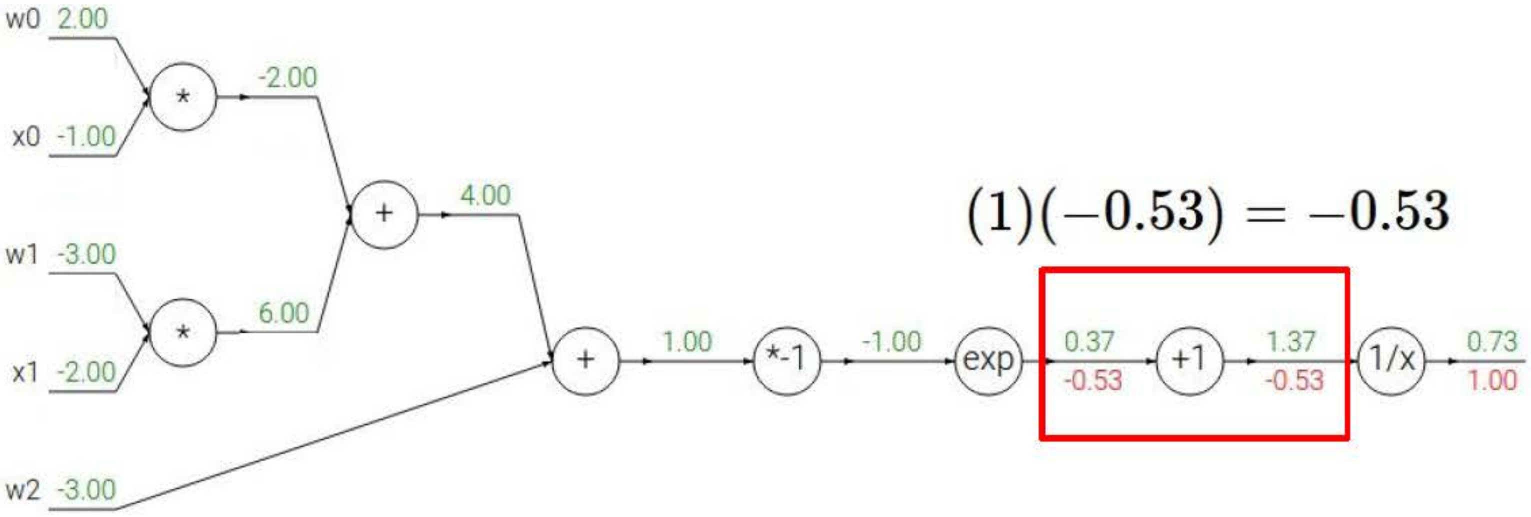
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$\boxed{f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1}$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$(1)(-0.53) = -0.53$$

$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

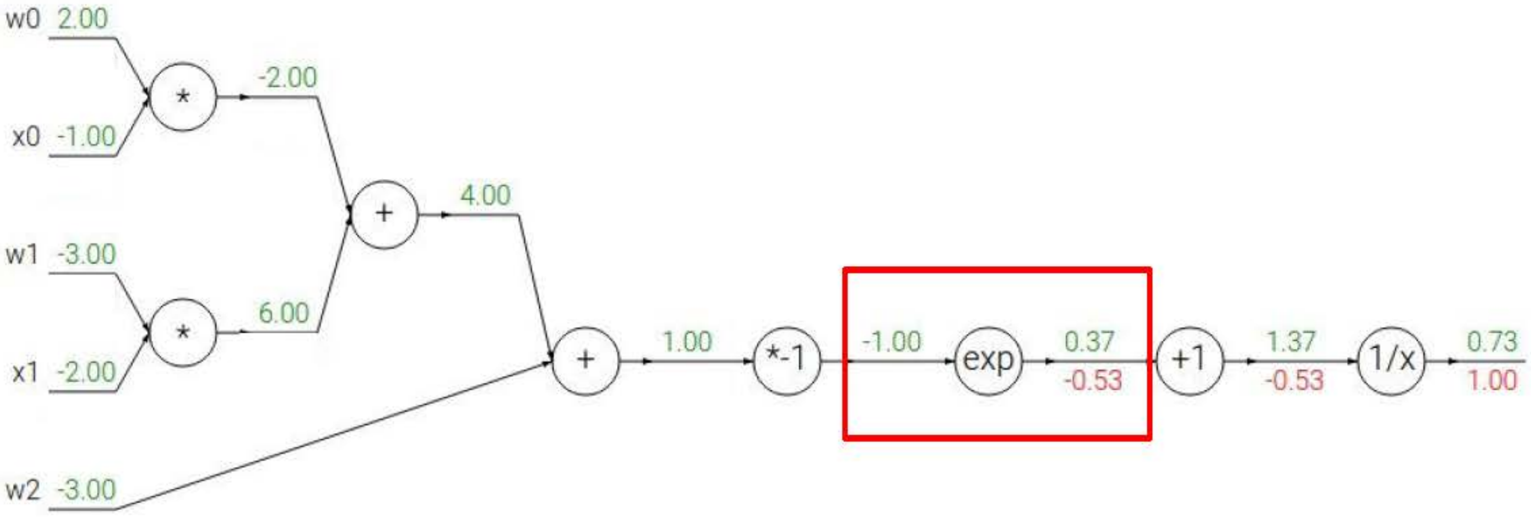
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$\boxed{f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1}$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

$$\boxed{f(x) = e^x \quad \to \quad \frac{df}{dx} = e^x}$$

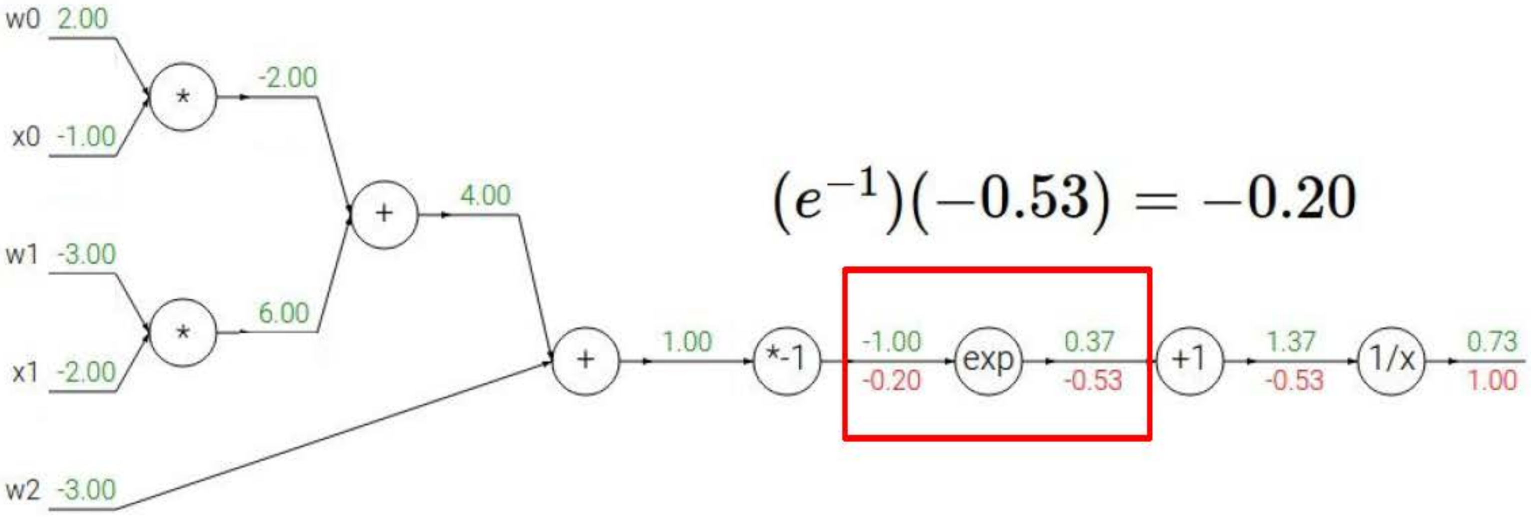
$$f_a(x) = ax \quad \to \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \to \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \to \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$(e^{-1})(-0.53) = -0.20$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

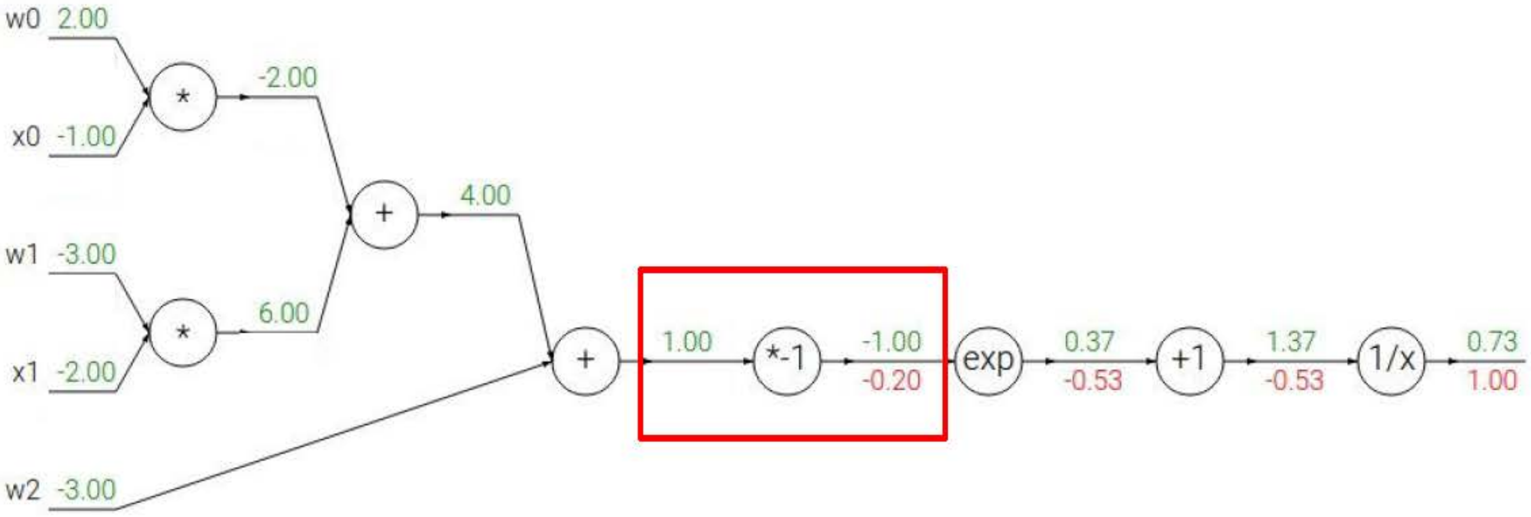
$$\boxed{f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x}$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

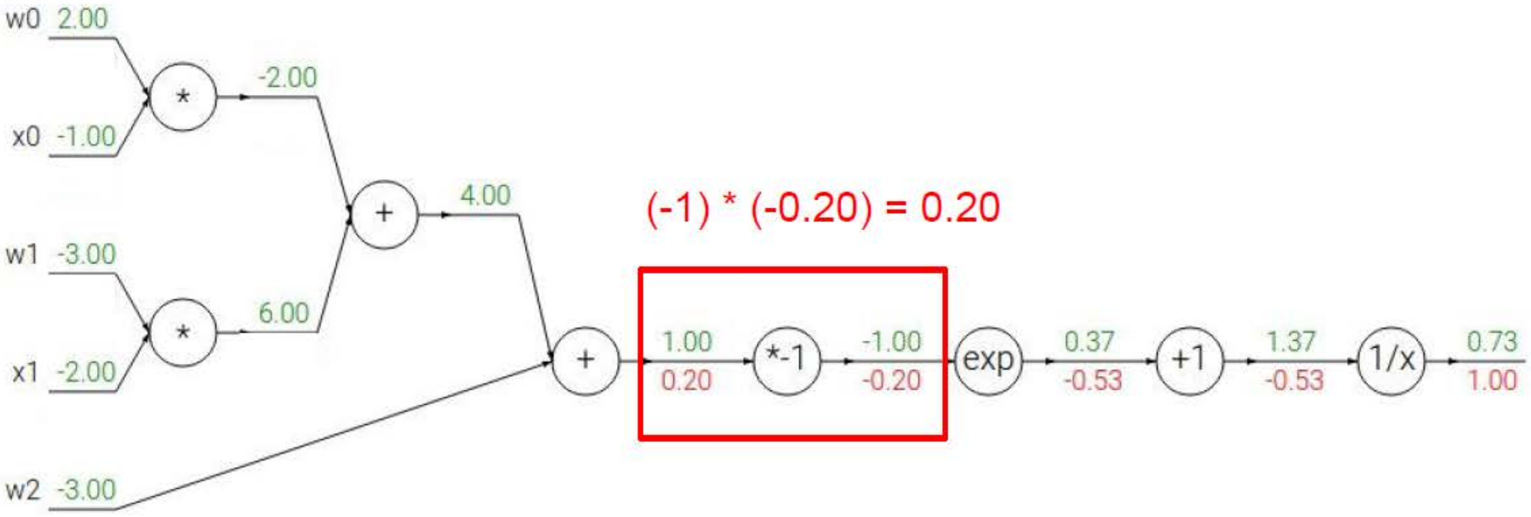
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f}\cdot\frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L}\cdot\frac{\partial a^L}{\partial w_j}$$

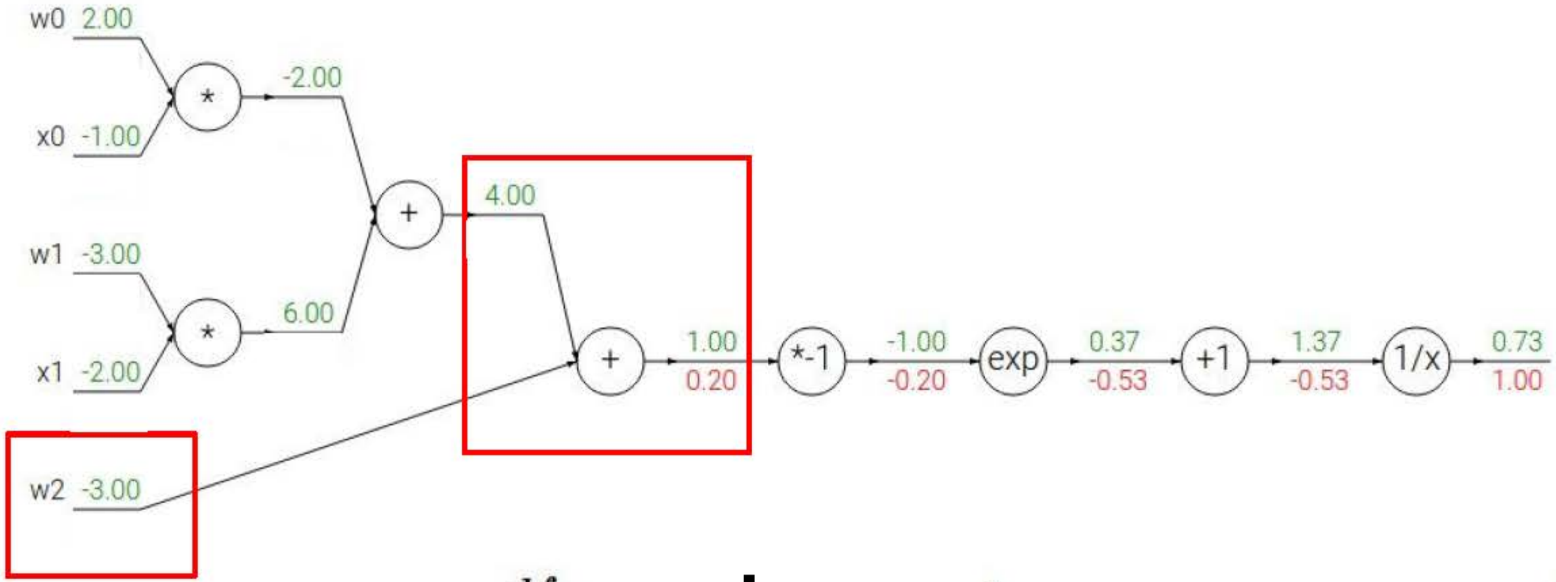
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$\boxed{f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a}$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

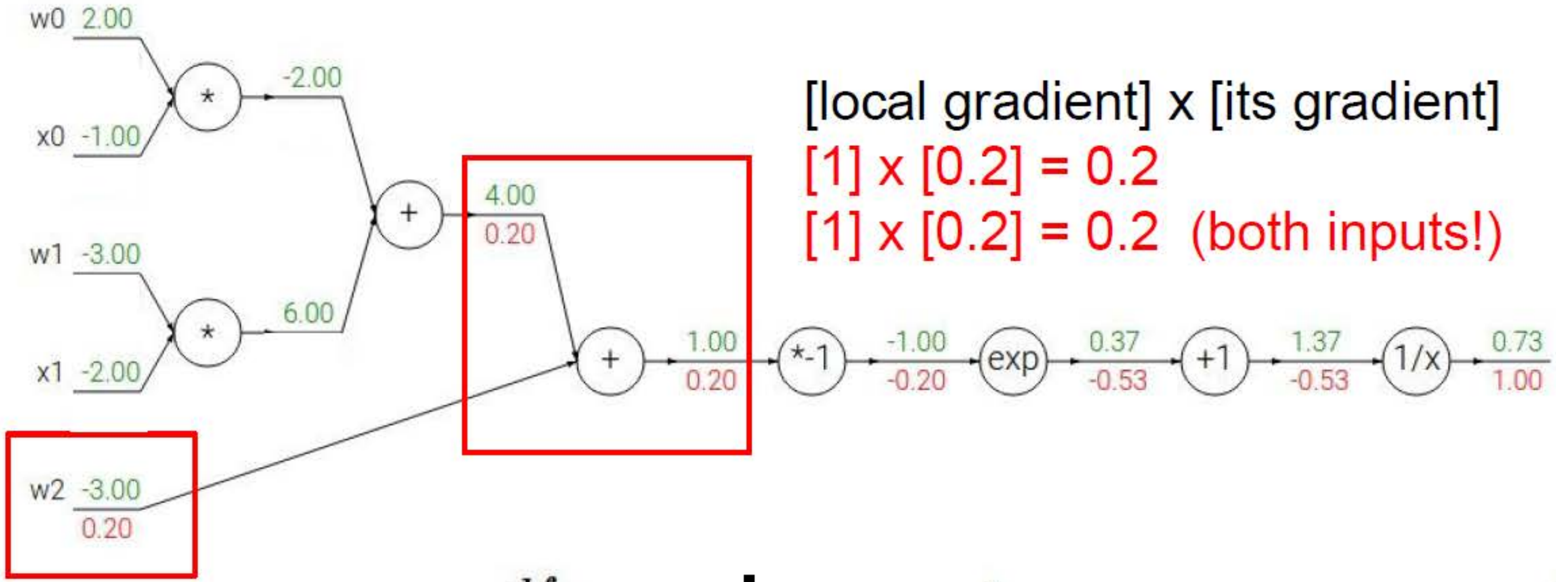
$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

[local gradient] x [its gradient]

[1] x [0.2] = 0.2

[1] x [0.2] = 0.2 (both inputs!)



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

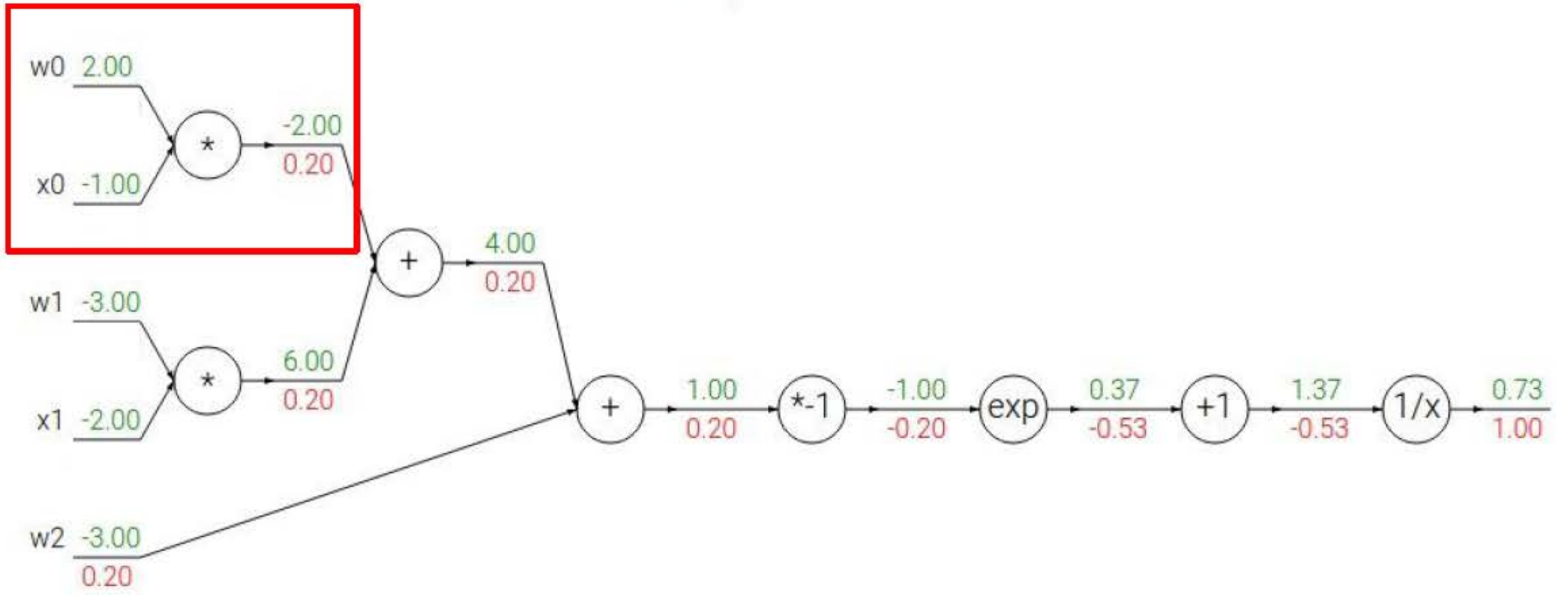
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0 x_0 + w_1 x_1 + w_2)}}$$



$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

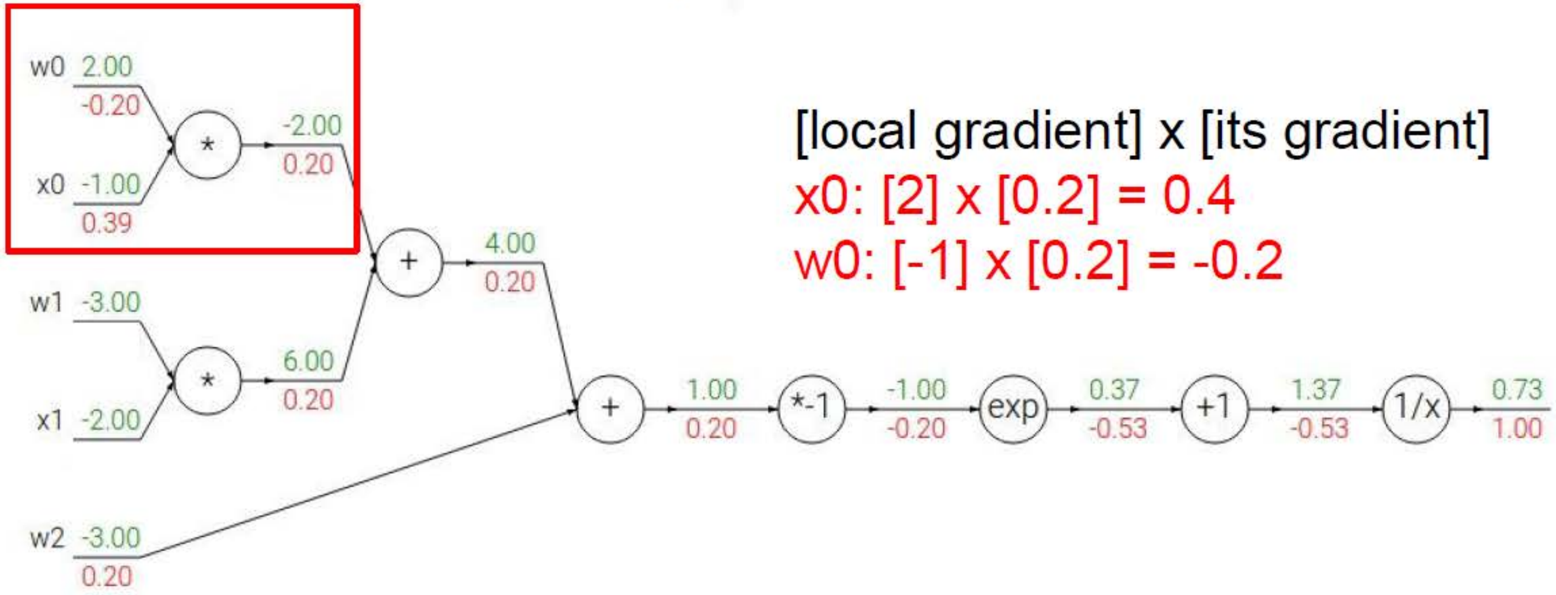
$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$



[local gradient] x [its gradient]

x0: [2] x [0.2] = 0.4

w0: [-1] x [0.2] = -0.2

$$\frac{\partial C}{\partial f} \cdot \frac{\partial f}{\partial w_j} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial w_j}$$

$$f(x) = e^x \quad \rightarrow \quad \frac{df}{dx} = e^x$$

$$f_a(x) = ax \quad \rightarrow \quad \frac{df}{dx} = a$$

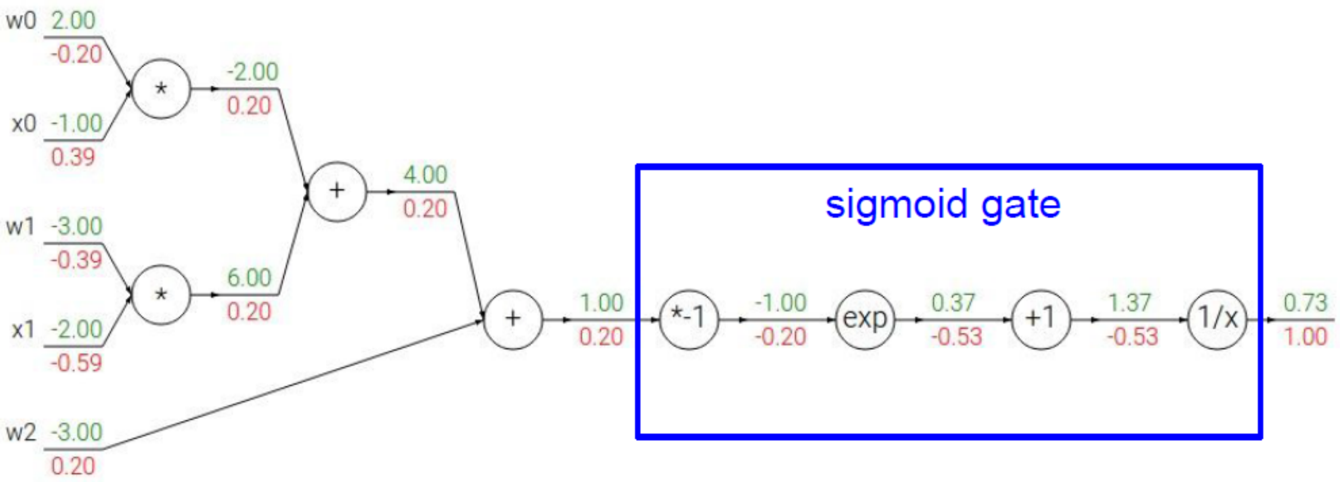
$$f(x) = \frac{1}{x} \quad \rightarrow \quad \frac{df}{dx} = -1/x^2$$

$$f_c(x) = c + x \quad \rightarrow \quad \frac{df}{dx} = 1$$

$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$\boxed{\sigma(x) = \frac{1}{1+e^{-x}}}$$ sigmoid function

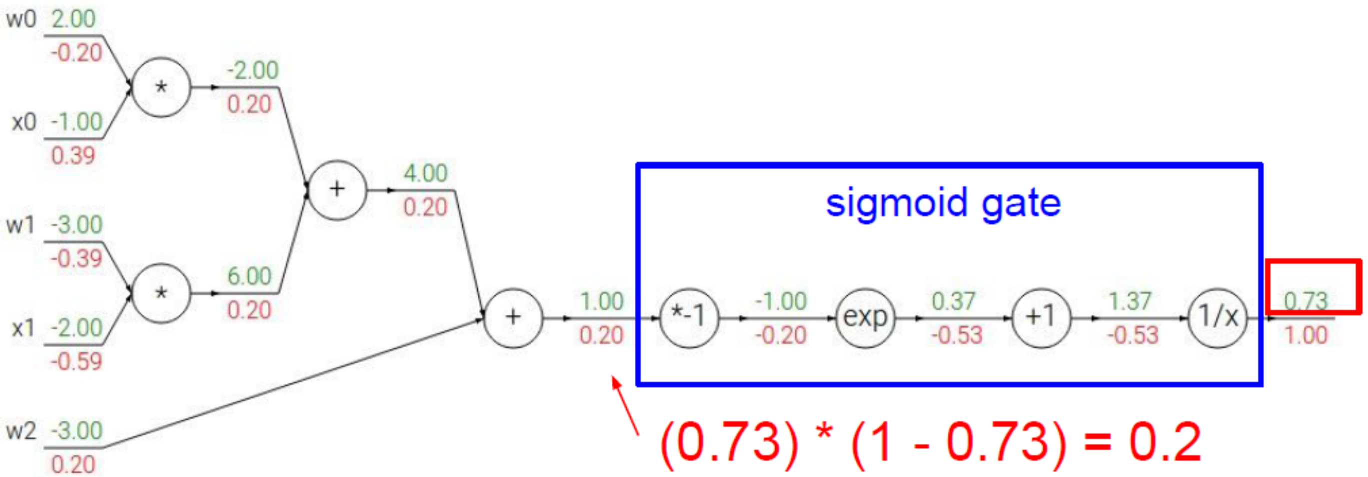
$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



$$f(w,x) = \frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

$$\boxed{\sigma(x) = \frac{1}{1+e^{-x}}}$$ sigmoid function

$$\frac{d\sigma(x)}{dx} = \frac{e^{-x}}{(1+e^{-x})^2} = \left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right) = (1-\sigma(x))\,\sigma(x)$$



Gradient update rule

$$w_j \leftarrow w_j - \eta \frac{\partial C}{\partial w_j}$$

Chain rule

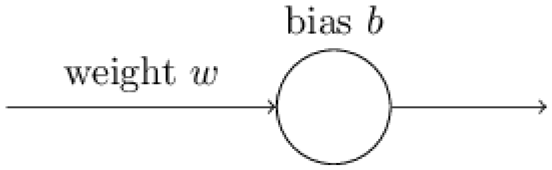
$$\frac{\partial C}{\partial w_j} = \frac{\partial C}{\partial f(w,x)} \frac{\partial f(w,x)}{\partial w_j}$$

More Loss Functions – Cross Entropy

- One Variable output

Cross-entropy loss function

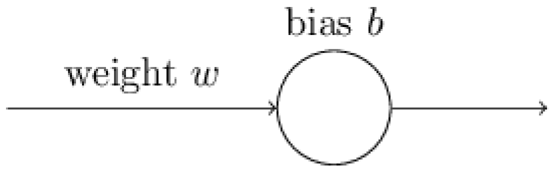
$$C = -\frac{1}{n} \sum_x \left[y \ln a + (1-y) \ln(1-a) \right],$$



Modeling *a* as the parameter of a Bernoulli r.v.

$$p(\widehat{y}) = (a^L)^y (1 - a^L)^{(1-y)}$$
$$y \in \{0, 1\}$$

Squared error loss function



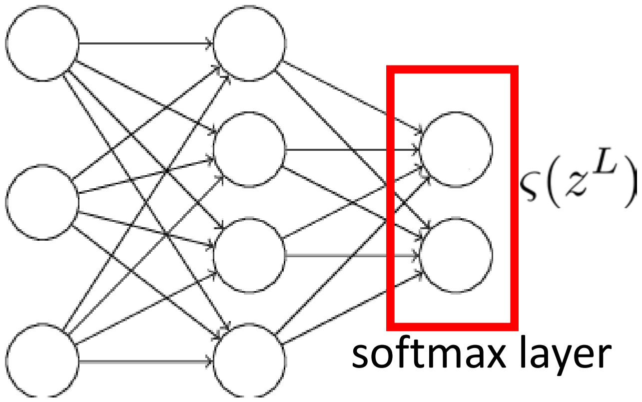
$$C = \frac{(y-a)^2}{2}$$

Cross Entropy – Multi-class via softmax

- Multi-variable output

Weighted input to softmax layer

$$z_j^L = \sum_k w_{jk}^L a_k^{L-1} + b_j^L$$



softmax layer activation

$$a_j^L(z_j) = \varsigma(z^L) = \frac{e^{z_j^L}}{\sum_k e^{z_k^L}}$$

$$\sum_j a_j^L = \frac{\sum_j e^{z_j^L}}{\sum_k e^{z_k^L}} = 1.$$

$$\begin{bmatrix} P(y=1|z^L) \\ \vdots \\ P(y=C|z^L) \end{bmatrix} = \begin{bmatrix} \varsigma(z^L)_1 \\ \vdots \\ \varsigma(z^L)_C \end{bmatrix} = \frac{1}{\sum_{k=1}^{C} e^{z_k^L}} \begin{bmatrix} e^{z_1^L} \\ \vdots \\ e^{z_C^L} \end{bmatrix}$$

Cross Entropy – Multi-class via softmax

- Multi-variable output

$$\underset{w}{\text{argmax}}\ \mathcal{L}(w; y, z^L, x) \qquad P(y, z|w, x) = P(y|z, w, x)P(z|w, x)$$

$$\begin{bmatrix} P(y=1|z^L) \\ \vdots \\ P(y=C|z^L) \end{bmatrix} = \begin{bmatrix} \varsigma(z^L)_1 \\ \vdots \\ \varsigma(z^L)_C \end{bmatrix} = \frac{1}{\sum_{k=1}^{C} e^{z_k^L}} \begin{bmatrix} e^{z_1^L} \\ \vdots \\ e^{z_C^L} \end{bmatrix}$$

For one forward pass/batch, w & x are fixed

$$\mathcal{L}(w; y, z^L) = P(y|z^L)$$

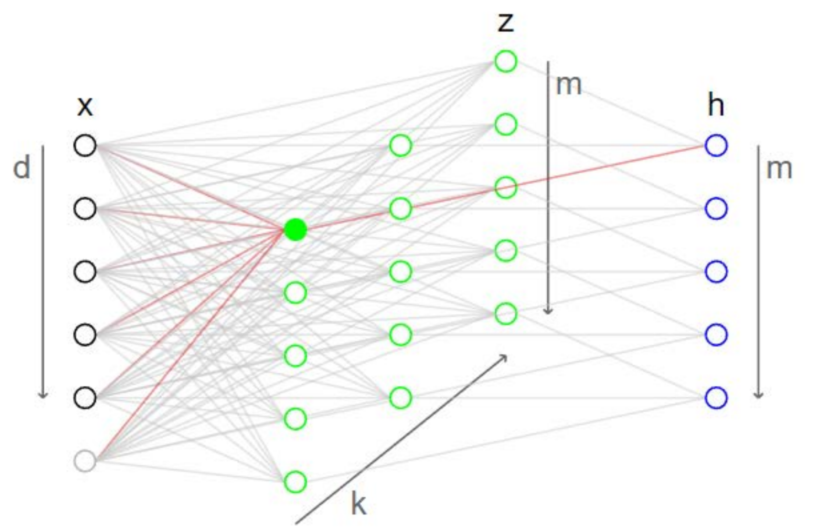
$$P(y|z^L) = \prod_{i=c}^{C} P(y_c|z^L)^{y_c} = \prod_{i=c}^{C} \varsigma(z^L)_c^{y_c} = \prod_{i=c}^{C} (a_c^L)^{y_c}$$

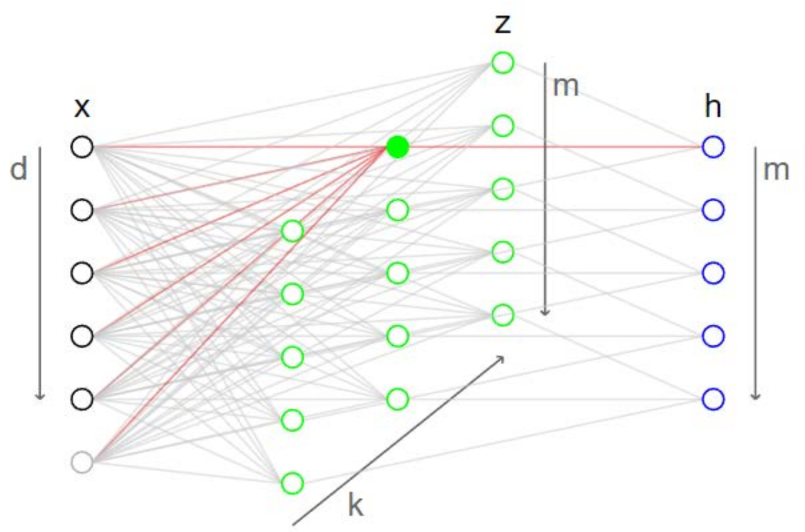
Training samples are (x,y) pairs, where y is a C-dimensional indicator vector, with C-1 elements zero.
The location of 1 indicates the class membership of point x

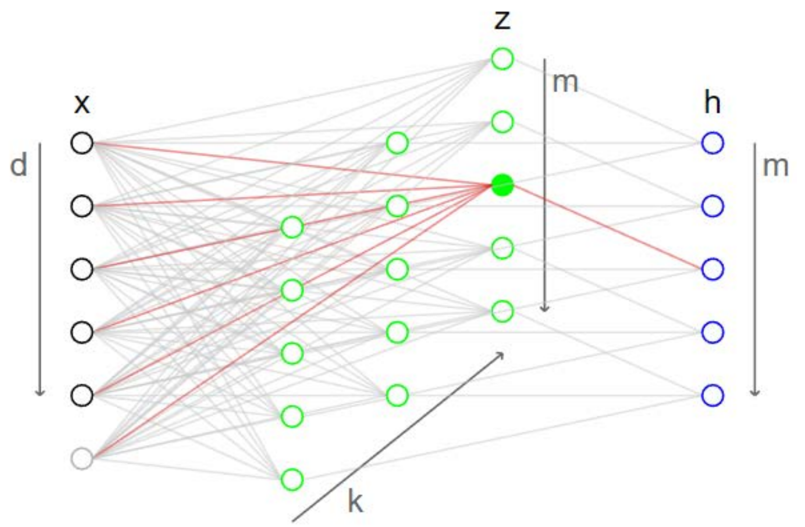
$$-log\mathcal{L}(w|y, z^L) = \xi(y, z^L) = -log\prod_{i=c}^{C} (a_c^L)^{y_c} = -\sum_{i=c}^{C} y_c \cdot log(a_c^L)$$

$$\xi(Y, A^L) = \sum_{i=1}^{n} \xi(y^{(i)}, a^{L^{(i)}}) = -\sum_{i=1}^{n}\sum_{i=c}^{C} y_c^{(i)} \cdot log((a_c^L)^{(i)})$$

Hidden Layers as Feature Representation

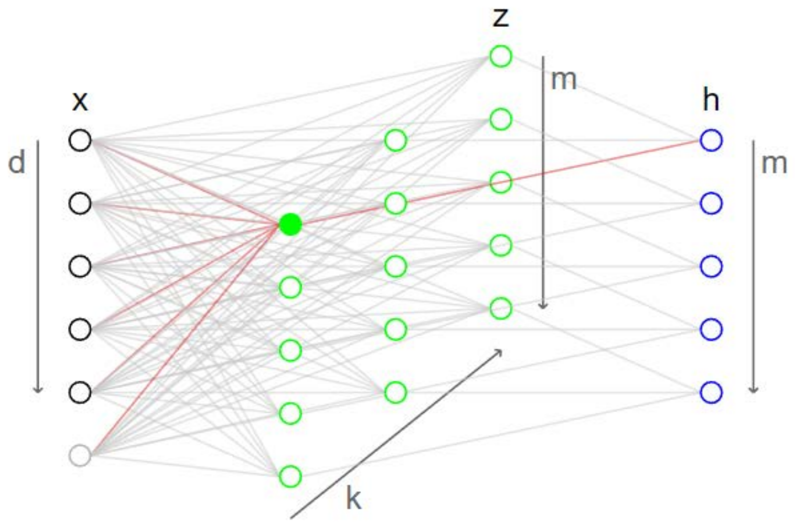


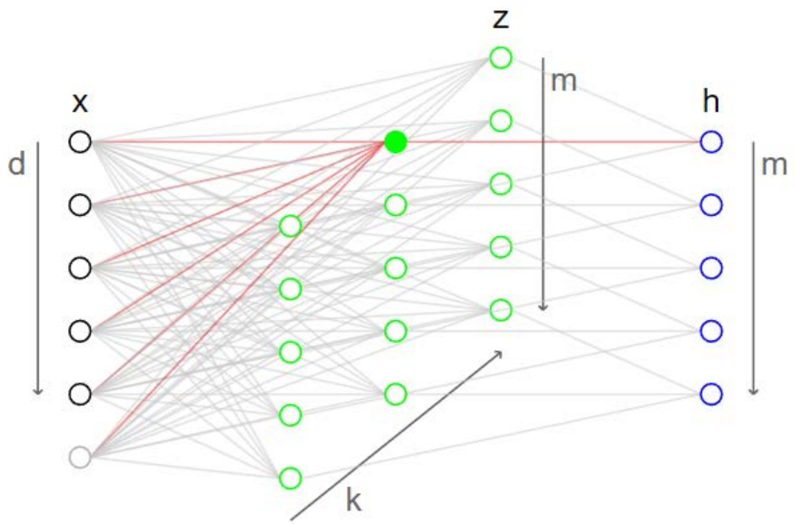


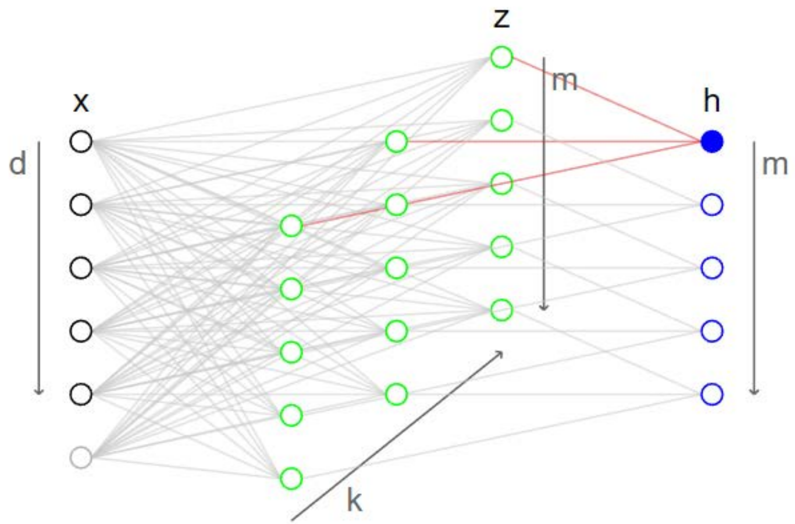


Hidden layers can be interpreted as
k different feature maps

Maxout Activation





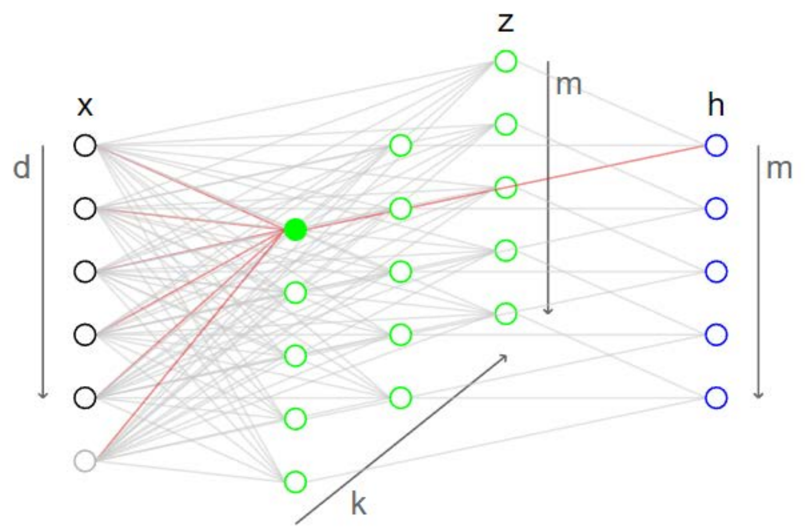


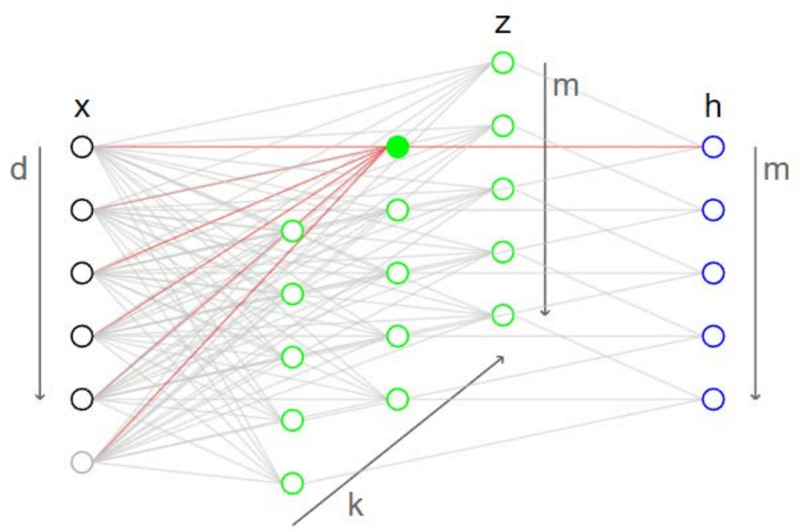
Hidden layers can be interpreted as
k different feature maps

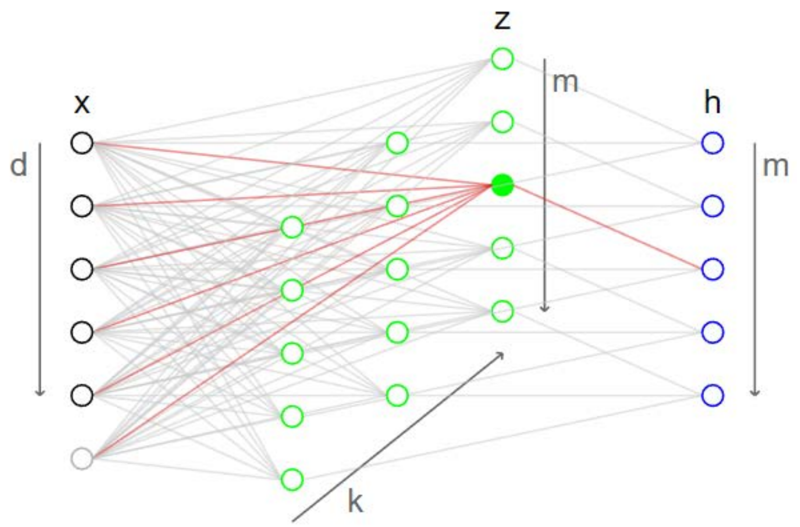
$$a_i(x) = \max_{j \in [1,k]} (z_{ij}), \quad i = 1, \dots, m$$

$$z_{ij} = x^\top w_{\dots ij} + b_{ij}$$

Convolutional Neural Networks - Motivation







Hidden layers can be interpreted as
k different feature maps

Multiple Feature Maps –> #parameters?

Practical Aspects

- SGD, Mini-Batch GD
- Regularization
- Initialization
- Normalization – Batch/Instance Normalization
- Shuffling of data
- Activation functions

Stochastic vs Batch Gradient

Advantages of Stochastic Learning

1. Stochastic learning is usually *much* faster than batch learning.
2. Stochastic learning also often results in better solutions.
3. Stochastic learning can be used for tracking changes.

Advantages of Batch Learning

1. Conditions of convergence are well understood.
2. Many acceleration techniques (e.g. conjugate gradient) only operate in batch learning.
3. Theoretical analysis of the weight dynamics and convergence rates are simpler.

Shuffling Examples

Choose Examples with Maximum Information Content

1. Shuffle the training set so that successive training examples never (rarely) belong to the same class.
2. Present input examples that produce a large error more frequently than examples that produce a small error.

Normalizing Inputs

Transforming the Inputs

1. The average of each input variable over the training set should be close to zero.
2. Scale input variables so that their covariances are about the same.
3. Input variables should be uncorrelated if possible.

Activation

Sigmoids

1. Symmetric sigmoids such as hyperbolic tangent often converge faster than the standard logistic function.
2. A recommended sigmoid [19] is: $f(x) = 1.7159 \tanh\left(\frac{2}{3}x\right)$. Since the tanh function is sometimes computationally expensive, an approximation of it by a ratio of polynomials can be used instead.
3. Sometimes it is helpful to add a small linear term, e.g. $f(x) = \tanh(x) + ax$ so as to avoid flat spots.

Or Use ReLUs or its leaky/parametric variants

Efficient Backprop by LeCun

Initialization

Initializing Weights

Assuming that:

1. the training set has been normalized, and
2. the sigmoid from Figure 1.4b has been used

then weights should be randomly drawn from a distribution (e.g. uniform) with mean zero and standard deviation

$$\sigma_w = m^{-1/2} \tag{1.16}$$

where m is the fan-in (the number of connections feeding *into* the node).

Learning rate

- Momentum

$$\Delta w(t+1) = \eta \frac{\partial E_{t+1}}{\partial w} + \mu \Delta w(t)$$

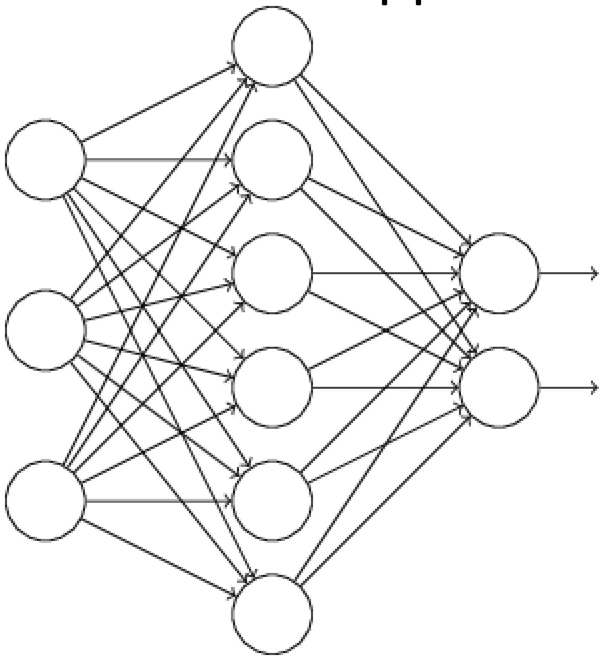
- Adaptive Learning Rates
  - Adagrad
  - Adam

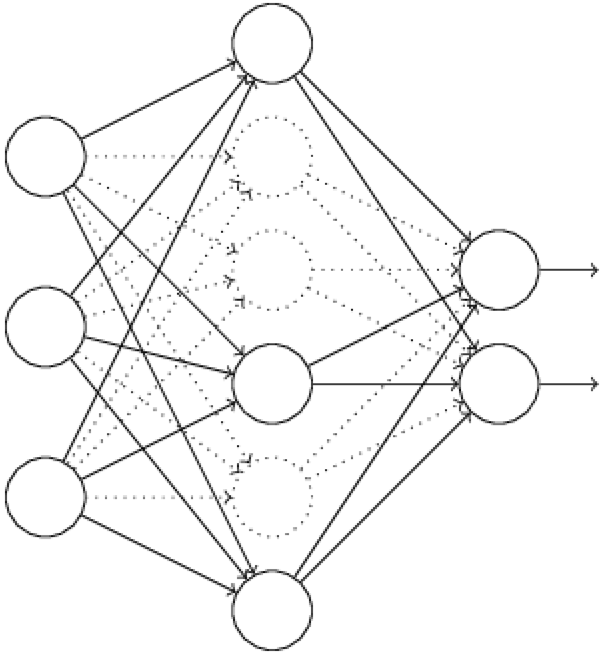
Equalize the Learning Speeds

- give each weight its own learning rate
- learning rates should be proportional to the square root of the number of inputs to the unit
- weights in lower layers should typically be larger than in the higher layers

Regularization

- L1 regularization
- L2 regularization (aka weight decay)
- Dropout
  - An ensemble approach





Why ‘Deep’ Networks?

- Compositional Structure
  - Each layer is a composition of ‘features’ from previous layer
  - Effective in learning a ‘hierarchy of concepts’